Velkommen til ellevte gruppetime!

Gruppe 8

Uke 6: skop (scope) og parameteroverføring



Dagens gruppetime

- **Første time:**
  - Hvordan løse en lang tekstoppgave?
  - Litt repetisjon av skop
  - Noen oppgaver

- **Andre time:**
  - Hva er og hvordan holder man styr på parameteroverføring?
  - Litt repetisjon av parameteroverføring
  - Noen oppgaver

Hvordan går man fram her? (1)

Skriv en funksjon felles som har en parameter tall_lister som tar inn en nøstet liste - en liste av lister av heltall. Funksjonen skal returnere en liste med alle heltall som finnes to eller flere ganger i den nøstede lista. Det er uten betydning om et slikt heltall finnes i flere ulike lister i den nøstede lista eller om det finnes flere ganger i den samme lista. Det er også uten betydning hvilken rekkefølge de felles tallene returneres i.

Kallet felles([[1,5,1], [3,4], [2,3]]) skal altså returnere enten [1,3] eller [3,1] siden tallene 1 og 3 finnes to steder i den nøstede lista (og siden rekkefølgen på tallene som returneres ikke er av betydning).

Hvordan går man fram her? (2)

Skriv en **funksjon felles** som har en **parameter tall_lister** som tar inn en **nøstet liste** - en liste av lister av **heltall**. Funksjonen skal **returnere en liste** med alle heltall som finnes **to eller flere** ganger i den nøstede lista. Det er **uten betydning om et slikt heltall finnes i flere ulike lister** i den nøstede lista eller om **det finnes flere ganger i den samme lista**. Det er også **uten betydning hvilken rekkefølge** de felles tallene returneres i.

Kallet felles([[1,5,1], [3,4], [2,3]]) skal altså returnere enten [1,3] eller [3,1] siden tallene 1 og 3 finnes to steder i den nøstede lista (og siden rekkefølgen på tallene som returneres ikke er av betydning).

Hvordan går man fram her? (3)

https://tinyurl.com/yserjrvc

Løsningsforslag (fra eksamenssensur)

```
def felles(tall_lister):
    noterte = []
    felles = []
    for liste in tall_lister:
        for tall in liste:
            if tall in noterte:
                if tall not in felles:
                    felles.append(tall)
            else:
                noterte.append(tall)
    return felles
```

Skop (scope)

- Variabler inni en funksjon er som regel ikke tilgjengelige i en annen funksjon
- "Skopet" handler om hvor i koden (filen) variablen er synlig eller tilgjengelig
- I IN1000 unngår vi "globale" variabler
  - Plutselig bruker man de i flere .py filer
- Flytt ytre kode i en funksjon (ofte kalt hovedprogram.py)

Oppgave 1

Ola, Anne og Kristian skal på telttur og har laget en felles liste over ting de må ha med. Før de drar, må de sjekke om de har husket det viktigste, telt og sovepose. Skriv en funksjon sjekk_og_pakk(pakkeliste) som tar inn en liste med strenger. Funksjonen skal sjekke om "Telt" og "Sovepose" finnes i listen. Hvis et av elementene mangler, skal funksjonen legge det til (endte) i listen. Funksjonen skal *ikke* returnere noe.

Eksempel: Hvis vi har listen ting = ["Kokeapparat", "Telt"] og kaller sjekk_og_pakk(ting), skal listen ting etterpå inneholde ["Kokeapparat", "Telt", "Sovepose"] (rekkefølgen er uten betydning).

Oppgave 2

Oskar og Espen skal sjekke menyene til ulike restauranter. Hver restaurant har en liste over allergener i sine retter, representert som en nøstet liste (en liste av lister med strenger).

Skriv en funksjon finn_trygge_restauranter(restauranter, allergi) som tar inn en nøstet liste over restauranter (hvor hver restaurant er en liste av retter, og hver rett er en liste av allergener) og en streng som representerer en allergi.

Funksjonen skal returnere en liste med indeksene til de restaurantene hvor absolutt ingen av rettene inneholder allergien.

```python
def modifiser_data(tall, liste):
    tall = tall + 10
    liste.append(99)
    liste = [1, 2, 3]
    return tall

mitt_tall = 5
min_liste = [10, 20]

nytt_tall = modifiser_data(mitt_tall, min_liste)

print(f"mitt_tall: {mitt_tall}")
print(f"nytt_tall: {nytt_tall}")
print(f"min_liste: {min_liste}")
```

Hva printes ut her?

Python Tutor forrige slide

Inneholder kode + kommentarer som forklarer hva som skjer

https://tinyurl.com/3yhe86vj

Parameteroverføring

- Parameteren tilordnes verdien av argumentet når funksjonen kalles
- "Argumentet" er verdien du sender inn, "parameteren" er variabelen som tar imot inni funksjonen
- Gjør at funksjoner kan ta imot data utenfra uten å bruke globale variabler
- I IN1000 skiller vi mellom overføring av uforanderlige verdier (som int og str) og foranderlige verdier (som lister, mengder)
  - val/var, mutable/immutable, PRIMITIVE datatyper (int, float, str, bool)
  - Endringer på en liste inni funksjonen vil ofte endre den originale listen på utsiden

Oppgave 3

Per, Palle og Espen lurer på om de skal finne på noe til kvelden. Dersom nøyaktig to av dem ønsker å spille badmington går de for det, men ellers (om færre enn to eller flere enn to vil dette) gjør de heller noe annet. Skriv en funksjon badmington(per_vil, palle_vil, espen_vil) som tar inn tre boolske argumenter, og returnerer True dersom to (og kun to) av de tre argumentene er True.

Altså skal for eksempel kallet badmington(True, True, False) returnere True, mens kallet badmington(True, True, True) skal returnere False

Oppgave 4

Espen programmerer et rollespill og holder styr på spillernes utstyr ved hjelp av en ordbok (dictionary). Hver spiller har et navn og en nøstet pakkeliste (en liste over gjenstander).

Skriv en funksjon oppgrader_utstyr(spiller_info, obligatorisk_gjenstand) som tar inn en ordbok spiller_info og en streng obligatorisk_gjenstand. Funksjonen skal sjekke om spilleren har gjenstanden i listen sin. Hvis spilleren mangler den, skal funksjonen legge den til i spillerens utstyrsliste. Funksjonen skal ikke returnere noe.

Ferdig?

- Tenk tilbake på ukene til nå, hva er noe du sliter med i IN1000?
  - Jobb med det,
  - Bruk filter på Trix

Løsningsforslag oppgave 1

```
def sjekk_og_pakk(pakkeliste):
    if "Telt" not in pakkeliste:
        pakkeliste.append("Telt")

    if "Sovepose" not in pakkeliste:
        pakkeliste.append("Sovepose")
```

Løsningsforslag oppgave 2

```
def finn_trygge_restauranter(restauranter, allergi):
    trygge_indekser = []
    for i in range(len(restauranter)):
        restaurant = restauranter[i]
        er_trygg = True

        for rett in restaurant:
            if allergi in rett:
                er_trygg = False

        if er_trygg:
            trygge_indekser.append(i)

    return trygge_indekser
```

Løsningsforslag oppgave 3

```
def badmington(per_vil, palle_vil, espen_vil):
    antall_vil = 0
    if per_vil:
        antall_vil += 1
    if palle_vil:
        antall_vil += 1
    if espen_vil:
        antall_vil += 1
    return antall_vil == 2
```

Løsningsforslag oppgave 4

```
def oppgrader_utstyr(spiller_info, obligatorisk_gjenstand):
    utstyrs_liste = spiller_info["utstyr"]

    if obligatorisk_gjenstand not in utstyrs_liste:
        utstyrs_liste.append(obligatorisk_gjenstand)
```